(19) Canadian Intellectual Property Office
An Agency of Industry Canada

Office de la Propri,t, Intellectuelle du Canada
Un organisme d'Industrie Canada

(11) CA 2 283 417 (13) A1

(40) 23.03.2001
(43) 23.03.2001

(12)

(21) 2 283 417

(51) Int. Cl.6: H04L 12/18, H04L 12/56

(22) 23.09.1999

(71) SPACEBRIDGE NETWORKS CORPORATION, 115 Champlain St., HULL, Q1 (CA).

(72) LEVESQUE, MARC (CA).
WIBOWO, EKO ADI (CA).
ANEESH, DALVI (CA).
AMAL, KHAILTASH (CA).
GILDERSON, JAMES A. (CA).

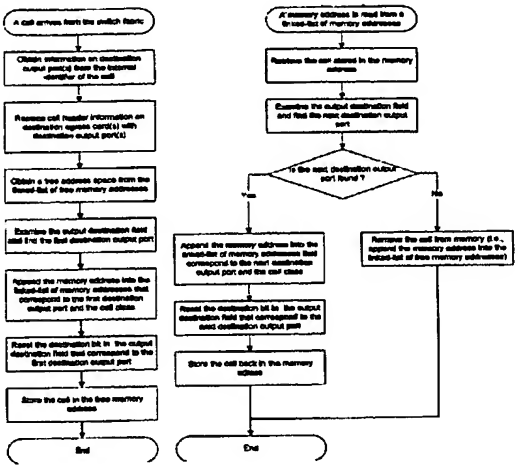
(74) BORDEN LADNER GERVAIS LLP

(54) ORDONNANCEMENT DE SORTIE ET GESTION DE TRAFIC DE SOURCES MULTIPLES DANS LES SYSTEMES DE COMMUTATION DES COMMUNICATIONS

(54) OUTPUT SCHEDULING AND HANDLING MULTICAST TRAFFIC IN COMMUNICATION SWITCHING SYSTEMS

(57)

A novel method and apparatus to support multicast communications are disclosed. The disclosed invention is based on recycling the multicast cell address as many times as necessary within the output card of a data switching system, which accommodates multiple output ports, thereby avoiding the complex processing required to generate copies of the cell or the cell address. The invention allows efficient and scalable implementation of multicast traffic scheduling at output ports/cards of a switch/router. These properties, combined with its low implementation requirement, render the invention particularly suitable for use in resource-constrained environment, such as satellite on-board packet switching systems.



OPIC
OFFICE DE LA PROPRIÉTÉ INTELLECTUELLE DU CANADA

CIPO
CANADIAN INTELLECTUAL PROPERTY OFFICE

(12)(19)(CA) **Demande-Application**

(21)(A1) **2,283,417**

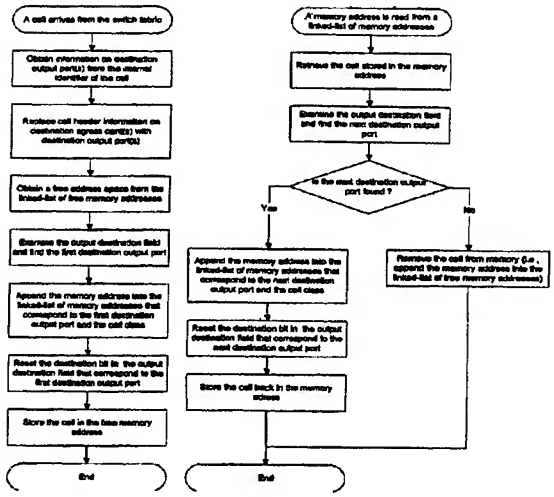
(22) 1999/09/23
(43) 2001/03/23

(72) GILDERSON, JAMES A., CA
(72) LEVESQUE, MARC, CA
(72) WIBOWO, EKO ADI, CA
(72) AMAL, KHAILTASH, CA
(72) ANEESH, DALVI, CA
(71) SPACEBRIDGE NETWORKS CORPORATION, CA

(51) Int.Cl.[6] H04L 12/18, H04L 12/56

(54) **ORDONNANCEMENT DE SORTIE ET GESTION DE TRAFIC DE SOURCES MULTIPLES DANS LES SYSTEMES DE COMMUTATION DES COMMUNICATIONS**

(54) **OUTPUT SCHEDULING AND HANDLING MULTICAST TRAFFIC IN COMMUNICATION SWITCHING SYSTEMS**



(57) A novel method and apparatus to support multicast communications are disclosed. The disclosed invention is based on recycling the multicast cell address as many times as necessary within the output card of a data switching system, which accommodates multiple output ports, thereby avoiding the complex processing required to generate copies of the cell or the cell address. The invention allows efficient and scalable implementation of multicast traffic scheduling at output ports/cards of a switch/router. These properties, combined with its low implementation requirement, render the invention particularly suitable for use in resource-constrained environment, such as satellite on-board packet switching systems.

ABSTRACT

A novel method and apparatus to support multicast communications are disclosed. The disclosed invention is based on recycling the multicast cell address as many times as necessary within the output card of a data switching system, which accommodates multiple output ports, thereby avoiding the complex processing required to generate copies of the cell or the cell address. The invention allows efficient and scalable implementation of multicast traffic scheduling at output ports/cards of a switch/router. These properties, combined with its low implementation requirement, render the invention particularly suitable for use in resource-constrained environment, such as satellite on-board packet switching systems.

TITLE OF INVENTION

Output Scheduling and Handling of Multicast Traffic in Data Switching Systems

FIELD OF APPLICATION

This invention relates to scheduling and handling of multicast traffic in an output interface card within Cell/packet switching systems, such as terrestrial and satellite switches or routers.

BACKGROUND OF THE INVENTION

The capability to support multicast communications is essential for a data communication switch. Current and future applications and services, such as video broadcasts, video-on-demand and teleconferencing, are dependent on the availability of multicast support. For the switch to support multicast communications and meet quality-of-service (QoS) levels expected by these applications often involves complex design issues, including the generation of multiple copies of cells (or packets), their internal routing to each destination and in some cases the maintenance of cell ordering. Examples of prior art methods for handling multicast communications are given in references [1] – [8].

In reference [1], a method of copy generation by recycling data cells in the switch fabric is disclosed. The method makes use of a copy-by-two network to create an additional data cell in each recycle. The recycling operation is then performed as many times as necessary to deliver the data cell to each of its destinations. The main strength of this multicast method is its simple and scalable technique for generating and routing each copy of a multicast data cell. Its main weakness, however, is its choice of where the cell-copy process is performed, *i.e.* in the switch fabric. If with this method a high throughput were to be maintained and congestion in the switch fabric were to be avoided while delaying multicast cells to generate their copies, the switch fabric would need to be made significantly faster than the aggregate ingress data rate and to be provided with sufficient buffer space. This makes the method

inappropriate for implementation in an environment where the speed and size of available Application-Specific Integrated Circuits (ASIC) are significant limiting factors, such as in satellite on-board switches.

A method disclosed in reference [7] moves the task of recycling multicast cells to the input/output ports. Similar to the method in reference [1], each multicast cell is recycled as many times as necessary to deliver it to all of its destinations. In reference [7], paths are setup between output ports and input ports and used to feedback each multicast cell that needs to be delivered to other destinations. In that sense, then, the copy generation process is performed recursively. This method shares the simplicity and design scalability advantages of the earlier one. Furthermore, by moving the task of recycling data cell from the switch fabric to the input/output ports, it avoids the above-mentioned limiting factors that can reduce throughput. The penalty in using this method for multicasting data cells, however, is the high delay that a multicast cell can experience. This drawback can potentially limit the application of this method for non-real-time applications only.

The methods disclosed in references [2] and [8] rely on a special copy network for generating copies of each multicast cell. This copy network is meant to be placed between the input ports and the switch fabric. Each multicast cell and its copies are then treated as unicast data cells. These methods do not require any design changes to the switch fabric or output ports and achieve predictable delay performance for each multicast cell. Disadvantages of these methods include the increased amount of buffer memory needed to store each copy of a multicast cell in the switch fabric and output ports and the additional complexity required to implement the copy network.

The methods disclosed in references [3] and [4] partially address the memory requirement shortcoming of methods disclosed in references [2] and [8] by utilizing a copy controller to generate copies of each multicast cell in output ports. Output ports, however, are still needed to store each copy of a multicast cell. In addition, the additional design requirement for implementing a copy controller is not trivial.

The methods disclosed in references [5] and [6] reduces further the memory requirement in output ports by only storing the original copy of each multicast cell and generating copies of each multicast cell address. Each copy of the multicast cell address is appended in a linked-list of multicast cell addresses for the destination output port. The copy generation and the queueing of cell addresses are either performed concurrently or at a speed much higher than the speed of multicast cell arrival. In addition, these methods consider the use of two separate networks for unicast and multicast traffic. As a result, they are not suitable for applications in resource-limited environment.

There is clearly a need for a novel multicast scheduling and handling method that addresses the problems mentioned with respect to the above prior art methods, by minimizing the system memory requirement and reducing the overall design complexity.

Prior art references include:

1. Jonathan S. Turner, US Patent #5402415: Multicast Virtual Circuit Switch using Cell Recycling.
2. Tony T. Lee, US Patent #4813038: Non-blocking Copy Network for Multicast Packet Switching.
3. Richard Barnett, US Patent #5436893: ATM Cell Switch Suitable for Multicast Switching.
4. Kiyoshi Aiki, et. al., US Patent #5410540: Shared-Buffer-Type ATM Switch Having Copy Function and Copy Method Thereof.
5. F. M. Chiussi, et. al., US Patent #5689505: Buffering of Multicast Cells in Switching Networks.
6. Guy Harriman and Yang-Meng Arthur Lin, US Patent #5898687: Arbitration Mechanism for a Multicast Logic Engine of a Switching Fabric Circuit.
7. Fabrizio Sestini, "Recursive Copy Generation for Multicast ATM Switching", IEEE/ACM Transactions on Networking, Vol. 5, No. 3, June 1997.

8. Wen De Zhong, et. al., "A Copy Network with Shared Buffers for Large-Scale Multicast ATM Switching", IEEE/ACM Transactions on Networking, Vol. 1, No. 2, April 1993.

SUMMARY OF THE INVENTION

Throughout this specification, the following acronyms and terminology will be used.

GLOSSARY OF ACRONYMS

ATM	Asynchronous Transfer Mode
CDV	Cell Delay Variation
CLR	Cell Loss Ratio
CTD	Cell Transfer Delay
ID	Identifier
IP	Internet Protocol
SCFQ	Self-Clocked Fair Queueing
VPI	Virtual Path Identifier
VCI	Virtual Channel Identifier
VSFQ	Virtual Time-Stamped Fair Queueing
WFQ	Weighted Fair Queueing

TERMINOLOGY

Ingress cards:	Input interface cards of the switch
Egress cards:	Output interface cards of the switch
Data unit:	Cell or packet
Data switching system:	Cell/packet switching system, including terrestrial and satellite switches and routers

This invention is concerned with the processing (handling and scheduling) of multicast traffic at the output side of a communication switching system. The invented method is based on recycling the multicast cell address within an output card

as many times as necessary to deliver the multicast cell to all output ports it is destined to. Only one copy of the multicast cell is stored in the output card's memory and only one copy of the multicast cell address is maintained. The method thus minimizes the memory requirement in output ports. Finally, the method is capable of handling unicast and multicast traffic in a uniform manner, removing the need to implement two separate module for unicast and multicast traffic and significantly reducing output port design complexity.

Therefore, in accordance with an aspect of this invention, there is provided a method for the scheduling and handling of multicast traffic in an output interface card applicable to a data switching system, comprising the following steps:

a) Modifying the header of each newly arriving cell from the switch fabric to include information on output port(s) it should be delivered to (see Figure 5 for the internal cell format in the egress card). The information is to be obtained from preferably an egress connection/flow table (see Figure 3 for the format of the egress connection/flow table), addressed by the internal cell identifier (see Figure 4 for the internal cell format in the ingress card).

b) Obtaining an unused/free address from a linked-list of free memory addresses.

c) Removing the address space from the linked-list of free memory addresses.

d) Appending the cell address to a linked-list of cell addresses corresponding to the first output port the cell is destined to, or the only output port it is destined to in the case of a unicast cell, and the cell class. This output port can be identified by finding the rightmost (or leftmost) destination bit in the destination field, which is located in the cell header, whose value is "1".

e) Storing the newly arriving cell in the unused/free memory.

f) Reading the contents of the linked-list of cell addresses one at a time according to the direction set by the traffic scheduler.

g) Retrieving the cell stored in the memory address obtained from the linked-list of cell addresses and delivering the stored cell to the corresponding output port.

h) Appending the cell address to a linked-list of cell addresses corresponding to the next output port the cell is destined to, if any.

i) Appending the cell address to the linked-list of free memory addresses if the cell has been delivered to all of its destination output ports. In effect, this step removes the stored cell from memory.

An embodiment of this invention, further comprises a method to maintain sequence ordering of multicast cells under multicast tree re-configuration, including the following steps:

a) Setting the value of the output destination field in the cell header of a cell from a connection to the value in the current destination field in the connection table of the corresponding connection when the cell arrives from the switch fabric (see Figure 10).

b) Incrementing the value of the counter field of a connection every time a cell from the corresponding connection arrives from the switch fabric (see Figure 10).

c) Decrementing the value of the counter field of a connection every time a cell from the corresponding connection is removed from memory (see Figure 10).

d) Updating the content of the current destination field of a connection when the value of the corresponding counter is zero (see Figure 10). The update process involves copying the whole content of the new destination field to the current destination field.

Alternatively, the method to maintain sequence ordering of multicast cells under multicast tree re-configuration includes the following steps:

a) Incrementing the value of the counter field of a connection every time a cell from the corresponding connection arrives from the switch fabric and the value of the new destination field is different from the value of the current destination field (see Figure 12).

b) Leaving the value of the counter field of a connection unchanged every time a cell from the corresponding connection arrives from the switch fabric and the value of the new destination field is the same as that of the current destination field (see Figure 12).

c) Resetting the value of the counter field of a connection to "0" every time the content of the current destination field of the corresponding connection is updated (see Figure 12).

d) Setting the value of the counter field in the cell header of a cell from a connection to the value of the counter field in the connection table of the corresponding connection when the cell arrives from the switch fabric (see Figure 12).

e) Setting the value of the output destination field in the cell header of a cell from a connection to the value in the current destination field in the connection table of the corresponding connection when the cell arrives from the switch fabric (see Figure 12).

f) Updating the content of the current destination field of a connection when the value of the counter field in the cell header of a cell from the corresponding connection is the same as that of the counter field in the connection table of the corresponding connection at the time the stored cell is removed from memory (see Figure 12). The update process involves copying the whole content of the new destination field to the current destination field.

g) Additionally, updating the content of the current destination field of a connection when the value of the counter field in the connection table of the corresponding connection has reached its maximum value (see Figure 12). The update process again involves copying the whole content of the new destination field to the current destination field.

Yet another embodiment of this invention comprises another method to maintain sequence ordering of multicast cells under multicast tree re-configuration including the following steps:

a) Incrementing the value of the counter field of a connection every time a cell from the corresponding connection arrives from the switch fabric and the value of the new destination field is different from the value of the current destination field (see Figure 12).

b) Leaving the value of the counter field of a connection unchanged every time a cell from the corresponding connection arrives from the switch fabric and the value of the new destination field is the same as that of the current destination field (see Figure 12).

c) Resetting the value of the counter field of a connection to "0" every time the content of the current destination field of the corresponding connection is updated (see Figure 12).

d) Setting the value of the counter field in the cell header of a cell from a connection to the value of the counter field in the connection table of the corresponding connection when the cell arrives from the switch fabric (see Figure 12).

e) Setting the value of the output destination field in the cell header of a cell from a connection to the value in the current destination field in the connection table of the corresponding connection when the cell arrives from the switch fabric (see Figure 12).

f) Updating the content of the current destination field of a connection when the value of the counter field in the cell header of a cell from the corresponding connection is the same as that of the counter field in the connection table of the corresponding connection at the time the stored cell is removed from memory (see Figure 12). The update process involves copying the whole content of the new destination field to the current destination field.

g) Updating the content of a destination bit in the current destination field of a connection when the value of the corresponding destination bit in the new destination field of the corresponding connection is "1" (see Figure 13). The update process involves copying the value of the corresponding destination bit in the new destination field to the destination bit in the current destination field.

h) Updating the content of a destination bit in the current destination field of a connection when the value of the counter field in the cell header of a cell from the connection is the same as that of the counter field in the connection table of the corresponding connection at the time the cell is delivered to the output port corresponding to the destination bit (see Figure 13). The update process involves copying the value of the corresponding destination bit in the new destination field to the destination bit in the current destination field.

i) Additionally, updating the content of the current destination field of a connection when the value of the counter field in the connection table of the corresponding connection has reached its maximum value (see Figure 13). The update process involves copying the whole content of the new destination field to the current destination field.

Another embodiment of this invention further comprises a method that lets multicast traffic to receive better treatment than unicast traffic in terms of the location of

memory address attachment in the linked-list of cell address, by further including the following steps:

a) Appending the cell address of a multicast cell to a given linked-list of cell addresses after the location pointed to by the corresponding multicast end pointer if at the time of appending the cell address the number of cell addresses in the given linked-list is equal to or exceeds the multicast attach threshold (see Figure 14).

b) Appending the cell address of a multicast cell to a given linked-list of cell addresses after the location pointed to by the corresponding end pointer if at the time of appending the cell address the number of cell addresses in the given linked-list is less than the multicast attach threshold (see Figure 14).

c) Appending the cell address of a unicast cell to a given linked-list of cell addresses after the location pointed to by the corresponding end pointer irrespective of the number of cell addresses in the given linked-list (see Figure 14).

Main advantages of the novel multicast handling and scheduling technique are as follows:

1. Efficient usage of memory.

   Unlike prior art schemes, the new multicast technique does not generate duplicate copies of each multicast cell or cell address. This brings about efficient memory usage, which is especially important for an environment where such resource is a limiting factor to the system's performance, such as in satellite on-board packet switches.

2. Low implementation complexity.

   Since there is no need to create and store copies of each multicast cell or cell address, the processing involved in performing these functions is avoided. As a result, implementation complexity is significantly reduced. This makes the scheme especially suitable for implementation in satellite on-board packet switches, where the speed and size of available Application Specific Integrated Circuit (ASIC) are limiting factors for implementation.

3. Scalable design.

The complexity of the multicast support circuitry is only slightly affected by the number of output ports. This significantly contributes to the scalability of access card design.

4. Simple and fair arbitration of unicast and multicast traffic.
   Since multicast traffic is queued into unicast traffic queue(s), there is no need to implement a separate traffic scheduler for multicast traffic. In addition to reducing the implementation complexity even further, this brings about fair treatment of multicast traffic with respect to unicast traffic.

BRIEF DESCRIPTION OF THE DRAWINGS

Exemplary embodiments of the invention will now be further described with references to the drawings in which same reference numerals designate similar parts throughout the figures thereof, and wherein:

Figure 1 illustrates in a block diagram an ATM 2-by-2 Switch;
Figure 2. illustrates a format of the information contained in the ingress connection/flow table in the switch shown in Figure 1;
Figure 3. illustrates a format of the information contained in the egress connection/flow table in the switch shown in Figure 1;
Figure 4. illustrates internal cell format used in the ingress card and switch fabric shown in Figure 1;
Figure 5. illustrates Internal cell format used in the egress card;
Figure 6. illustrates the flow of a multicast cell in the egress card;
Figure 7. illustrates in a flowchart the multicast handling method in accordance with this invention;
Figure 8. illustrates a situation leading to out-of-order transmission of multicast cells;
Figure 9. illustrates egress connection table format (sequence ordering important);
Figure 10. illustrates in a flow chart a first update method ;
Figure 11. illustrates internal cell format used in the egress card (sequence ordering important, Scheme 2);
Figure 12. illustrates in a flow chart a second update method;
Figure 13. illustrates in a flow chart a third update method;

Figure 14. illustrates in a flow chart the cell address attachment method;

Table 15. summarizes simulation scenarios for Simulation Sets 1, 2 and 3;

Table 16. summarizes simulation scenarios for Simulation Set 4;

Table 17. summarizes simulation scenarios for Simulation Set 5;

Table 18. summarizes simulation scenarios for Simulation Set 6;

Table 19. summarizes buffer sizes and traffic scheduler weights for CBR and rt-VBR traffic in the fabric element

Table 20. summarizes buffer sizes and traffic scheduler weights for CBR and rt-VBR traffic in the egress card

Figure 21. shows peak-to-peak CDV of multicast CBR traffic for Simulation Set 1, Scenarios 1-6;

Figure 22. shows peak-to-peak CDV of multicast CBR traffic for Simulation Set 1, Scenarios 7-12;

Figure 23. shows peak-to-peak CDV of multicast CBR traffic for Simulation Set 1, Scenarios 13-18;

Figure 24. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 2, Scenarios 1-6;

Figure 25. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 2, Scenarios 7-12;

Figure 26. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 2, Scenarios 13-18;

Figure 27. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 3, Scenarios 1-6;

Figure 28. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 3, Scenarios 7-12;

Figure 29. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 3, Scenarios 13-18;

Figure 30. shows peak-to-peak CDV of multicast CBR traffic for Simulation Set 4;

Figure 31. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 5; and

Figure 32. shows peak-to-peak CDV of multicast rt-VBR traffic for Simulation Set 6.

DETAILED DESCRIPTION OF THE INVENTION

I. Overall Switch Multicast Handling

Before a description on the overall switch multicast handling method is presented, a brief explanation of the embodied traffic queueing and traffic scheduling strategies as well as the switch fabric is provided. This invention, however, is by no means dependent on the use of these strategies or the type of switch fabric described herein. Furthermore, descriptions of the invention are made with reference to Asynchronous Transfer Mode (ATM) terminology and ATM switching equipment, but the scope of the invention includes other data switching systems besides ATM.

A switch can generally be broken down into 3 main elements, namely input (ingress) interface cards, the switch fabric and output (egress) interface cards (in some switches, an interface card normally functions as both input and output interface cards). Figure 1 depicts in a block diagram of an ATM 2-by-2 switch. An ingress card is responsible for processing both unicast and multicast traffic from input ports for internal routing and other purposes. With respect to routing, additional information is appended in the header portion of the ATM cell. Such additional information typically includes the internal identifier (ID) of the ATM cell, the cell type, the cell class and the destination egress card(s). This information is stored in a table that is indexed by the external ID of the ATM cell, which is the combination of the Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) of the cell. Figure 2 shows some information contained in the ingress connection/flow table. The ingress card is considered to be able to accept traffic from 8 input ports. Note that depending on the number of egress cards, it may be preferable to have information regarding the destination egress cards stored in memory instead of appended in the cell header. The location where such information is stored, however, is not particularly relevant to the main concept behind this invention.

Next, the switch fabric is responsible for multiplexing traffic from all ingress cards and forwarding traffic to all egress cards. A single-stage, crossbar-type switch fabric

is considered. The switch fabric is composed of several small switching sub-elements, which will be referred here as fabric element. Each fabric element accepts traffic from two ingress cards and forwards traffic to one egress card.

Finally, the egress card is responsible for processing traffic received from the switch fabric for transmission outside the switch through output ports. This processing involves removing information appended in the cell header by the ingress card and appending new information for delivering each cell to one or several output ports. This information is stored in a table that is indexed by the internal ID of the cell. Figure 3 shows some information contained in the egress connection/flow table. The egress card is also considered to be able to accommodate 8 output ports.

Within the switch, traffic is always queued according to its service class. In the ingress card, unicast and multicast traffic is queued differently. The ingress card treats unicast traffic differently than it does multicast traffic. Virtual output queueing strategy is only employed for unicast traffic. That is, unicast traffic is also queued according to its destination egress card. The ingress card does not explicitly perform multicast cell replication. Instead, each multicast cell is broadcast to the switch fabric. Then, all fabric elements that handle traffic for those output cards the multicast cell is destined to perform implicit copying of the cell.

The switch fabric is capable of recognizing to which egress card(s) a cell is destined. A special internal cell format is used to achieve this capability. As shown in Figure 4, a multicast indicator bit in the internal cell format is used to differentiate among unicast and multicast cells. All multicast cells have this bit set to "1", whereas all unicast cells have this bit set to "0". Furthermore, a destination field in the internal cell format is used to indicate to which egress card(s) the cell should be delivered. This field is composed of several bits, where each one is concerned with a particular egress card. The bit value is set to "1" if the cell is to be delivered to the particular egress card. Thus, a multicast cell may have more than one bit whose values are set to "1". A unicast cell, on the other hand, can only have one bit whose value is set to "1".

In the switch fabric, traffic is also queued according to its source ingress card. The switch fabric does not perform any multiple cell replications for multicast purpose. The switch fabric treats unicast and multicast traffic uniformly.

Finally, in the egress card, traffic is again queued according to its destination egress port. In contrast to prior art multicast handling methods, in an embodiment of this invention, the egress card treats unicast and multicast traffic in a similar way. The egress card does not need to perform explicit cell replication for multicast purpose and uses a slightly different internal cell format than that used in the ingress card or in the switch fabric. The cell format is shown in Figure 5. As shown in this figure, the multicast indicator bit is still retained for the purpose of differentiating among unicast and multicast cells. The destination field, however, is now used to indicate to which output port(s) the cell is destined. Similarly, this field is composed of several bits, where each one is concerned with a particular output port. The bit value is set to "1" if the cell is to be delivered to the particular output port. Thus, a multicast cell may have more than one bit whose values are set to "1", whereas a unicast cell can only have one bit whose value is set to "1". Note that depending on the number of output ports within an egress card, it may be preferable to have information regarding the destination output ports stored in memory instead of appended in the cell header. The location where such information is stored, however, is not particularly relevant to the main concept behind this invention.

At last, a distributed traffic scheduling strategy is employed throughout the switch. A traffic scheduler is located in each element of the switch, including each fabric element.

II. Multicast Traffic Handling in Output Interface Cards

Instead of having the egress card perform explicit cell replication for multicast purpose as in prior art methods, a multicast cell is delivered to all output ports it is destined to using the technique described below, in accordance with this invention.

Each cell newly received by the egress card will have its header information modified to identify which output port(s) it should be delivered to. As explained above, this information is obtained from the egress connection/flow table, which is addressed by the internal cell identifier. An unused/free address is then obtained from a linked-list of free memory addresses and is used to store the cell. The cell address will then be appended to a linked-list of cell addresses corresponding to the first output port the cell is destined to, or the only output port it is destined to in the case of a unicast cell, and the cell class. The first destination output port can be obtained by examining the destination field located in the cell header. As explained above, each bit in the destination field corresponds to a particular output port. Therefore, the rightmost bit, *i.e.* the least significant bit, in the destination field whose value is "1" can be used to identify the first destination output port. Alternatively, the leftmost bit, *i.e.* the most significant bit, whose value is "1" can also be used to identify the first destination output port to receive the cell. Note that the cell type, i.e., unicast or multicast, does not need to be used to determine which linked-list the cell address should be appended to, only the content of the destination field and the information regarding the cell class are. That is, the method handles both unicast and multicast traffic in a unified manner.

The contents of this linked-list is then read one at a time according to the direction set by the traffic scheduler. A cell address that is read from the linked-list will be used to retrieve the stored cell, which is then delivered to the corresponding output port. At this stage, the destination field of the cell header is re-examined to see if the stored cell should be delivered to other output ports. If the stored cell needs to be delivered to one or more output ports, the cell address is then appended to the linked-list of cell addresses corresponding to the next output port and the cell class. If, however, the stored cell has been delivered to all of its destination output ports, the cell will be removed from memory and the memory address will be returned / appended to the linked-list of free memory addresses.

Figure 6 depicts the process of delivering a multicast cell to output ports it is destined to. Figure 7 describes the flowchart of multicast (and unicast) handling method.

As described above, the multicast handling method allow unicast and multicast cells to receive similar treatment in terms of traffic scheduling through the common process of destination field manipulation. As a result, the need of a special process and a separate module for handling multicast traffic can be avoided. This advantage ultimately allows the design of the egress card to scale well with the number of output ports it needs to accommodate. There is, however, a performance implication to multicast traffic transmission delay as the number of output ports increases.

III. Preserving Cell Ordering under Multicast Tree Reconfiguration

In some connection-based networks, such as ATM networks, the preservation of cell sequence ordering is of importance. The multicast handling method described above in accordance with this invention, may not be able to maintain the sequence ordering of multicast cells of a connection under all circumstances. Specifically, multicast cells of a connection can be transmitted out-of-order for a short period of time when the multicast tree is reconfigured such that within an egress card, a leaf that is not the last leaf in the egress card is disconnected. Figure 8 depicts the situation that may lead to out-of-order transmission of multicast cells.

A number of methods to mitigate this difficulty have been conceived. These methods update the content of the destination field when it is considered safe to do so. These update methods require additional information on the new multicast destinations to be associated to each connection, preferably in the form of a new field in the egress connection table. This additional field, henceforth referred to as the new destination field, is used to store the new multicast destinations, is initialized to the original multicast destinations when the multicast connection is setup, and is updated whenever the multicast tree is reconfigured. The initialization and update of the new destination field is normally instructed by the connection control process, which is outside the scope of this invention.

The existing destination field, henceforth referred to as the current destination field, is also initialized to the original multicast destinations when the multicast connection is setup. The initialization of the current destination field is also normally instructed by the connection control process. The update of the current destination field, however, is for most cases only allowed when it is safe to do so according to the update method. A special case exists in which the connection control process can directly update the current destination field. This special case is when the update is due only to the addition of one or more destinations. The content of the current destination field is updated with the partial or whole content of the new destination field, depending on the employed update method. Figure 9 depicts the new format of the egress connection table.

This first update method requires that per-VC accounting be performed in the egress card. Per-VC accounting refers to the practice of counting the number of cells from a connection that is currently stored in memory. In order to perform per-VC accounting, a counter needs to be associated to each connection, preferably in the form of a new field in the egress connection table. The value of this additional field is incremented when a new cell from the corresponding connection arrives from the switch fabric, and is decremented when a stored cell of the corresponding connection is removed from memory. The first update method allows the update of the current destination field when the value of this counter / field is zero. The update process involves copying the whole content of the new destination field to the current destination field. Figure 10 shows the flowchart of the first update method.

The first update method is the simplest among the three update methods described in this section. The other two update methods, however, offer more in terms of reliability and performance. More specifically, these methods are more resistant to the impact of bit error to the cell counter used in the first method. Furthermore, the third method allows updates to be performed sooner than the first and second methods do.

The second update method also requires that a counter be associated to each connection, preferably in the form of a new field in the egress connection table. The value of this counter, however, is manipulated differently. The counter is incremented every time a new cell from the corresponding connection arrives from the switch fabric, provided the value of the new destination field is different from the value of the current destination field. On the other hand, the counter is reset to "0" after the content of the current destination field is updated, and is not changed when the value of the new destination field is the same as that of the current destination field.

Furthermore, the second update method requires that an additional information be incorporated into each cell, preferably in the form of a new field in the internal cell header. The content of this field is then set to the value of the counter when the cell arrives from the switch fabric. The content of this field will be used to determine when it is safe to update the value of the current destination field. In this respect, the second update method allows the update of the current destination field when the value of the new field in the cell header is the same as that of the counter field in the connection table at the time the stored cell is removed from memory. Additionally, in order to ensure that an update will be performed eventually, even with the risk of mis-ordering, the update can be performed when the maximum value of the counter field has been reached. Similar to the first method, the update process involves copying the whole content of the new destination field to the current destination field. Figure 11 depicts the internal cell format in the egress card to allow the use of the second update method. Figure 12 shows the flowchart of the second update method.

The third update method requires that the same information be associated to each connection and to each cell as that required by the second method. The difference between the second and the third method is in the method of updating the current destination field. In the third update method, the update of the current destination field is performed partially, bit-by-bit. Furthermore, there are two conditions in which the third update method allows the update of a given bit in the current destination field. First, the bit value can be updated when the value of the

corresponding bit in the new destination field is "1", since there is no risk of cell mis-ordering when a new destination is added. Second, the bit value can be updated when the value of the new field in the cell header is the same as that of the counter field in the connection table at the time the stored cell is delivered to the output port corresponding to the given bit. Finally, similar to the second method, the update can also be performed when the maximum value of the counter has been reached. Figure 13 shows the flowchart of the third update method.

IV. Improving Delay Performance of Multicast Traffic

Since each multicast cell address is recycled from the linked-list of its first destination output port to the linked-list of its last destination output port (within an egress card), the delivery delay that a multicast cell undergoes to its last destination output port is the sum of the delivery delay that it undergoes in its other destination output ports. Thus, one may consider that the multicast handling method would delay delivery of multicast cells excessively to certain output ports. Performance evaluation results as given further below show that the multicast handling method performs well in this respect under reasonable traffic loads.

Nevertheless, methods for improving delay performance of multicast traffic are described herein, since these methods also improve the performance of each of the three update methods described earlier. This is because by improving the delay performance of multicast traffic, the updates of the current destination field can be performed sooner. As a result, the risk of mis-ordering multicast cells due to counter wrap-around (the counter resets to 0 once incremented past its maximum value) is reduced.

Improving delay performance of multicast traffic can be achieved if multicast traffic receive better treatment than unicast traffic in terms of the location of memory address attachment in the linked-list of cell addresses. A number of alternative embodiments can achieve this objective. The method described below should serve well to illustrate the idea.

The cell address attachment method makes use of an additional pointer, henceforth referred to as multicast end pointer, to the linked list (normally, two pointers are associated to a linked list, a start pointer and a tail/end pointer). Next, the method uses the following per-class threshold to determine where in the linked-list of cell addresses a multicast cell address is to be appended / inserted:

- multicast attach threshold
  Indicate the linked-list size threshold above which the multicast end pointer will be used to indicate where the address of a multicast cell will be appended / inserted.

Then, if the number of cell addresses in a given linked-list is equal to or exceeds the list's corresponding threshold, the cell address of a multicast cell will be inserted after the location pointed to by the corresponding multicast end pointer. If the number of cell addresses in a given linked-list is less than the list's corresponding threshold, however, the cell address of a multicast cell will be inserted after the location pointed to by the end pointer. The cell address of a unicast cell is always inserted after the location pointed to by the end pointer. Figure 14 shows the flowchart of the cell address attachment method.

V. Performance Evaluations

Performance evaluations of the multicast handling method in accordance with this invention are performed through software modeling and simulation of a switching system. The objectives of these performance evaluations are to demonstrate that buffer memory can be efficiently utilized and cell delay variation (CDV) can be maintained low. The methods of improving delay performance of multicast traffic described in the previous section are not simulated in order to show that the multicast handling method by itself performs well in terms of maintaining low CDV values under reasonable traffic loads.

V.1 Resource Management Strategies

Resource management strategies employed in each simulation include traffic prioritization, traffic queueing, buffer memory partitioning, traffic scheduling and congestion control strategies. The employed traffic queueing strategy has been explained above. The rest of the employed resource management strategies are described below.

As described earlier, a traffic scheduler is located in each element of the switch, including each fabric element. A variant of the Weighted Fair Queueing (WFQ) scheme called the Virtual Time-Stamped Fair Queueing (VSFQ) scheme is employed to perform this traffic scheduling function. This scheme is a virtual clock version of the well-known Self-Clocked Fair Queueing (SCFQ) scheme, as described in *"S. J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Applications", In Proceedings of IEEE Infocom, June, 1994."*

For the purpose of these simulations, traffic is categorized into four service classes, namely Constant Bit Rate (CBR), real-time Variable Bit Rate (rt-VBR), non-real-time Variable Bit Rate (nrt-VBR) and Unspecified Bit Rate (UBR) service classes, and into three priority classes, namely High Priority, Medium Priority and Low Priority classes. CBR traffic belongs to the High Priority class and receives higher service precedence than any other traffic. Next, rt-VBR traffic belongs to the Medium Priority class and receives higher service precedence than other non-real-time traffic. Finally, non-real-time traffic (i.e., nrt-VBR and UBR) belongs to the Low Priority class. As a result, non-real-time traffic has lower service precedence than real-time traffic. Traffic prioritization is employed in the ingress card and switch fabric, but not in the egress card.

Furthermore, in order to make efficient use of buffer memory and achieve predictable cell loss and delay performance, a strategy of partial buffer sharing is employed in the ingress and egress cards. A strategy of complete buffer partitioning is employed in the switch fabric, however. In the ingress and egress cards, each service class is allocated a certain amount of dedicated buffer memory to store traffic. Furthermore,

each traffic queue of a given service class is also dedicated a certain amount of dedicated buffer memory. Then, all traffic queues of a given service class share an amount of shared buffer memory equal to the amount of dedicated buffer memory of the corresponding service class minus the total amount of dedicated buffer memory allocated to these traffic queues. Each traffic queue in the switch fabric is allocated a certain amount of buffer memory.

Finally, the overall congestion control strategy consists of fabric backpressure strategy, selective cell discard strategy for real-time traffic and selective packet discard strategy for non-real-time traffic. Fabric backpressure strategy is employed to eliminate congestion loss of non-real-time traffic in the buffer-limited switch fabric. Selective cell discard strategy is employed to reduce congestion loss of high priority real-time traffic by discarding low priority real-time traffic when congestion is detected. Selective packet discard is employed to increase buffer utilization and throughput of non-real-time traffic by storing complete packets only.

V2. Simulation Scenarios

Six sets of simulations are conducted. In each simulation, a traffic source can generate both unicast and multicast traffic. Multicast traffic generated by a traffic source, however, is always intra-egress. This simply means that all destination output ports of each multicast cell are located in one egress card. Therefore, the switch fabric does not need to implicitly replicate any multicast cell.

The first set of simulation scenarios involves only CBR traffic. This simulation set consists of 18 simulation scenarios. In each simulation scenario, the average traffic load placed at each egress card is set to 90%. This traffic load is generated by a number of constant-rate sources that each transmits cells at a rate of 414 cells/second. Next, the proportion of multicast traffic to unicast traffic is varied, with the objective of assessing the impact of different unicast traffic loading situation to delay performance of multicast traffic. Finally, the number of output port destinations (i.e., fan-out) of each multicast cell is also varied in order to observe any delay

performance degradation as the number of multicast cell fan-out is increased. Table 15 summarizes each simulation scenario for the first simulation set.

The second set of simulation scenarios involves both CBR and rt-VBR traffic. This simulation set also consists of 18 simulation scenarios. In each simulation scenario, the average traffic load placed at each egress card is set to 50%, of which CBR and rt-VBR traffic loads both correspond to 25%. CBR traffic is generated by the same type of sources used in the first simulation set, whereas rt-VBR traffic is generated by VBR II sources. Next, for the same reasons as previously described, the proportion of multicast traffic to unicast traffic and the number of multicast cell fan-out are varied. Table 15 summarizes each simulation scenario for the second simulation set. Note that VBR II traffic sources are generally used for switch benchmarking purposes. Their parameters are specified in Bellcore GR-1110-CORE specifications.

The third set of simulation scenarios is similar to the second set of simulation scenarios. The difference between these two sets of simulations is that in the third simulation set, rt-VBR traffic is generated by higher-speed VBR I sources. Table 15 summarizes each simulation scenario for the third simulation set. Note that VBR I traffic sources are also generally used for switch benchmarking purposes. Their parameters are specified in Bellcore GR-1110-CORE specifications.

The fourth set of simulation scenarios involves only CBR traffic. This simulation set consists of 5 simulation scenarios. The load of CBR traffic at each egress card is increased, from 50% in the first simulation scenario to 92.5% in the fifth simulation scenario. This is achieved by increasing the number of CBR sources in each input port to produce the desired traffic load. In each of these simulation scenarios, multicast traffic load accounts to approximately half of the load placed on each egress card. Furthermore, each multicast cell is destined to all 8 output ports of an egress card. Table 16 summarizes each simulation scenario for the fourth simulation set.

The fifth set of simulation scenarios involves both CBR and rt-VBR traffic and VBR II traffic sources are used to generate rt-VBR traffic. This simulation set also consists

of 5 simulation scenarios. In each simulation scenario, the load of CBR traffic at each egress card is maintained at 25%. The load of rt-VBR traffic at each egress card, however, is increased, from 25% in the first simulation scenario to 62.5% in the fifth simulation scenario. This is again achieved by increasing the number of VBR II traffic sources in each input port to produce the desired traffic load. Furthermore, similar to the fourth simulation set, multicast traffic load accounts to approximately half of the load placed on each egress card and each multicast cell is destined to all 8 output ports of an egress card. Table 17 summarizes each simulation scenario for the fifth simulation set.

The sixth set of simulation scenarios is similar to the fifth set of simulation scenarios. The difference between the fifth and the sixth simulation sets is that in the sixth simulation set, rt-VBR traffic is generated by higher-speed VBR I traffic sources. Table 18 summarizes each simulation scenario for the sixth simulation set.

V3. Simulation Parameters

Simulations are conducted for an 8-by-8 (i.e., 8 ingress and 8 egress cards) switch with crossbar-type switching fabric at the ingress/egress speed of 124 Mbps. Cell loss ratio (CLR) values are collected at each element of the switch (i.e., ingress card, switch fabric and egress card). Cell transfer delay (CTD) statistics are also collected at the output of the switch. Traffic destination is uniformly distributed into all 64 output ports. The simulated time for each simulation run is 10 seconds. CLR and CTD statistics are collected when the simulated time has exceeded 1.0 second.

A buffer size of 32768 cells is considered for both the ingress and egress cards, and a buffer size of 512 cells is considered for each fabric element.

Table 19 summarizes buffer sizes and traffic scheduler weights for CBR and rt-VBR traffic in the fabric element. As can be seen from the table, CBR and rt-VBR traffic is each allocated a buffer space of 64 cells. The buffer space dedicated to each service class is then divided equally between the 2 traffic buffers (recall that each fabric

element accepts traffic from 2 ingress cards; traffic arriving from one ingress card is buffered separately from traffic arriving from the other ingress card). Finally, CBR traffic from one ingress card receives the same level of service from the traffic scheduler as CBR traffic from the other ingress card (i.e., the scheduling weights for both traffic are the same). Similarly, rt-VBR traffic from one ingress card receives the same level of service from the traffic scheduler as rt-VBR traffic from the other ingress card.

Table 20 summarizes buffer sizes and traffic scheduler weights for CBR and rt-VBR traffic in egress card. As can be seen from the table, CBR traffic is allocated a total buffer space of 1280 cells. Next, rt-VBR traffic is allocated a total buffer space of 1280 cells for simulation sets 2 and 5, and a total buffer space of 5120 cells for simulation sets 3 and 6. The buffer space dedicated to each service class is then distributed to unicast and multicast traffic buffers (of the corresponding service class) such that the dedicated buffer space for each of the 8 unicast traffic buffers (one unicast traffic buffer for each output port) is the same, and that the dedicated buffer space for the multicast traffic buffer is twice that for each unicast traffic buffer. Next, except for simulation sets 3 and 6, CBR traffic receives the same level of service from the traffic scheduler as rt-VBR traffic. In simulation sets 3 and 6, CBR traffic is receiving a higher level of service from the traffic scheduler than rt-VBR traffic.

Of course, numerous variations and adaptations may be made to the particular embodiments of the invention described above, without departing from the spirit and scope of the invention, which is defined in the claims.

V4. Simulation Results

Cell loss of rt-VBR traffic due to congestion is noticed only in scenarios 4 and 5 of simulation set 5. These two simulations involve both CBR and rt-VBR traffic. In scenario 4, rt-VBR traffic load is approximately 58%, whereas in scenario 5, it is approximately 62.5%. Assuming that the effective cell rate of each VBR II source is approximately 320% higher than its average cell rate, and that 25% of the egress

capacity is consumed by CBR traffic (the effective cell rate of each constant-bit-rate source is considered to be the same as its average cell rate), these traffic loads as percentages of the egress capacity correspond to approximately
25% + 58% * (1 + 3.2) = 270% and
25% + 62.5% * (1 + 3.2) = 288%, respectively.
Thus, it can be concluded that the multicast handling method performs well in terms of buffer memory utilization.

Next, Figure 21, Figure 22 and Figure 23 show the peak-to-peak CDV values for output port 7 in simulation set 1. Note that in all simulations, the first destination output port is identified by finding the rightmost bit (i.e., least-significant bit) in the destination field whose value is "1". As a result, the peak-to-peak CDV value for output port 7 will be the worst among all output ports.

Figure 21 show simulation results for the case where the fan-out of each multicast CBR cell is 2. Figure 22 show simulation results for the case where the fan-out of each multicast CBR cell is 4. Finally, Figure 23 show simulation results where the fan-out of each multicast CBR cell is 8 (i.e., each cell is broadcast to all output ports). As shown in these figures, CDV values are well below 400 μsec and most are well below 300 μsec. Considering the high traffic load of 90% in each of these simulations, these results show that the multicast handling method performs well in maintaining low CDV values.

Figures 24, 25 and 26 show the peak-to-peak CDV values of rt-VBR traffic for output port 7 in simulation set 2, for the case where the fan-out of each multicast cell is 2, 4 and 8, respectively. As shown in these figures, CDV values are well below 200 μsec. Recall that the total traffic load in each simulation is 50%, of which 25% comes from rt-VBR traffic. Then, considering that the effective cell rate of each VBR II source is approximately 320% higher than its average cell rate, the total traffic load as a percentage of the egress capacity corresponds to approximately
25% + 25% * (1 + 3.2) = 130%.

These results show that the multicast handling method performs well in maintaining low CDV values, even for variable-bit-rate traffic sources.

Next, Figures 27, 28 and 29 show the peak-to-peak CDV values of rt-VBR traffic for output port 7 in simulation set 3, for the case where the fan-out of each multicast cell is 2, 4 and 8, respectively. Recall that each simulation in simulation set 3 involves higher-speed variable-bit-rate traffic sources (i.e., VBR I traffic sources). Correspondingly, each rt-VBR traffic queue is allocated a larger buffer space in order to avoid congestion loss. Therefore, the delay performance that the switch can offer to both unicast and multicast rt-VBR traffic will be reduced. Next, similar to simulation set 2, the total traffic load in each simulation is 50%, of which 25% comes from rt-VBR traffic. Then, assuming that the effective cell rate of each VBR I source is approximately 280% higher than its average cell rate, the total traffic load as a percentage of the egress capacity corresponds to approximately
$25\% + 25\% * (1 + 2.8) = 120\%$.

Finally, as shown in Figures 27, 28 and 29, CDV values range from slightly less than 300 μsec to slightly more than 500 μsec. These results confirm those of simulation sets 1 and 2.

Figure 30 presents the results of simulation set 4. Recall that in this simulation set, each multicast cell is to be delivered to all output ports of an egress card it is destined to. Figure 30 shows that the peak-to-peak CDV value for output port 7 increases more rapidly as traffic load increases. Nevertheless, the peak-to-peak CDV value for output port 7 is still below 250 μsec at a traffic load of 87.5%.

Figure 31 presents the results of simulation set 5. In this simulation set, each multicast cell is also to be delivered to all output ports of an egress card it is destined to. Figure 31 shows a dramatic increase of the peak-to-peak CDV value for output port 7 as traffic load increases. The peak-to-peak CDV value at 63% total traffic load, however, is still reasonable (approximately 200 μsec). This traffic load as a percentage of the egress capacity corresponds to approximately

25% + 38% * (1 + 3.2) = 185%.

Figure 32 presents the results of the last simulation set. Recall that simulation set 6 is similar to simulation set 5, except in the type of traffic source representing rt-VBR traffic (i.e., VBR I traffic sources are used in simulation set 6, whereas VBR II traffic sources are used in simulation set 5). Figure 32 also shows that the peak-to-peak CDV value increases steeply as traffic load increases. It can be estimated from that figure that at a traffic load of 80%, the peak-to-peak CDV value is approximately 2.5 msec. This traffic load as a percentage of the egress capacity corresponds to approximately
25% + 55% * (1 + 2.8) = 234%.

CLAIMS

What is claimed is:

1. A method of processing multicast traffic in a data switching system having an output interface card with a memory for receiving cells from a switch fabric, each cell having a header, said method comprising the steps of:

a) modifying the header of each newly received cell from the switch fabric, by inserting information on at least one output port that is to receive said newly received cell;

b) obtaining an unused address from a linked-list of free memory addresses for use as a cell address for the newly received cell;

c) removing an address space corresponding to the unused address from said linked-list of free memory addresses;

d) appending the cell address to a linked-list of cell addresses corresponding to the at least one output port that is to receive the newly received cell, and to the cell class.

e) storing the newly received cell in an unused memory portion corresponding to the unused memory address;

f) reading contents of the linked-list of cell addresses one at a time;

g) retrieving the cell stored in the memory address obtained from the linked-list of cell addresses and delivering the stored cell to the corresponding output port;

h) in case there is a next output port that is to receive the newly received cell, appending the cell address to a linked-list of cell addresses corresponding to said next output port; and

i) appending the cell address to the linked-list of free memory addresses if the cell has been delivered to all output ports that are to receive the newly received cell, thereby removing the stored cell from memory.

2. A traffic processing method as in claim 1, wherein said information is obtained from an egress connection table addressed by an internal cell identifier.

3. A traffic processing method as in claim 2, further comprising a method of maintaining sequence ordering of multicast cells under multicast tree re-configuration.

4. A traffic processing method as in claim 3, wherein said method of maintaining sequence ordering of multicast cells comprises the steps of:
a) setting an output destination field value in the cell header of a cell from a connection to the current destination field value in the connection table of the corresponding connection when the cell is received from the switch fabric;
b) incrementing the counter field value for a connection every time a cell from the corresponding connection is received from the switch fabric;
c) decrementing the counter field value for a connection every time a cell from the corresponding connection is removed from the memory; and
d) updating the content of the current destination field of a connection when the value of the corresponding counter is zero.

5. A traffic processing method as in claim 4, wherein said updating step includes copying the entire content of the new destination field to the current destination field.

6. A traffic processing method as in claim 3, wherein said method of maintaining sequence ordering of multicast cells comprises the steps of:
a) incrementing the counter field value of a connection every time a cell from the corresponding connection is received from the switch fabric and the new destination field value is different from the current destination field value.
b) leaving the counter field value for a connection unchanged every time a cell from the corresponding connection is received from the switch fabric and the new destination field value is similar to the current destination field value;
c) resetting the counter field value for a connection to "0" every time the current destination field content for the corresponding connection is updated;
d) setting the counter field value in the cell header of a cell from a connection to the counter field value in the connection table of the corresponding connection when the cell is received from the switch fabric;

e) setting the output destination field value in the cell header of a cell from a connection to the current destination field value in the connection table of the corresponding connection when the cell is received from the switch fabric;
f) updating the current destination field content for a connection when the counter field value in the cell header of a cell from the corresponding connection is similar to the counter field value in the connection table of the corresponding connection at the time the stored cell is removed from the memory; and
g) further updating the current destination field content of a connection when the counter field value in the connection table of the corresponding connection has reached a maximum value.

7. A traffic processing method as in claim 6, wherein each of the two updating steps includes copying the entire content of the new destination field to the current destination field.

8. A traffic processing method as in claim 3, wherein said method of maintaining sequence ordering of multicast cells comprises the steps of:
a) incrementing the counter field value for a connection every time a cell from the corresponding connection is received from the switch fabric and the new destination field value is different from the current destination field value;
b) leaving the counter field value for a connection unchanged every time a cell from the corresponding connection is received from the switch fabric and the new destination field value is similar to the current destination field value;
c) resetting the counter field value for a connection to "0" every time the current destination field content of the corresponding connection is updated;
d) setting the counter field value in the cell header of a cell from a connection to the counter field value in the connection table of the corresponding connection when the cell is received from the switch fabric;
e) setting the output destination field value in the cell header of a cell from a connection to the current destination field value in the connection table of the corresponding connection when the cell is received from the switch fabric;

f) updating the current destination field content for a connection when the counter field value in the cell header of a cell from the corresponding connection is similar to the counter field value in the connection table of the corresponding connection at the time the stored cell is removed from the memory;

g) updating the content of a destination bit in the current destination field of a connection when the corresponding destination bit value in the new destination field of the corresponding connection is "1";

h) updating the content of a destination bit in the current destination field of a connection when the counter field value in the cell header of a cell from the connection is similar to the counter field value in the connection table of the corresponding connection when the cell is delivered to the output port corresponding to the destination bit;

i) further updating the current destination field content for a connection when the counter field value in the connection table of the corresponding connection has reached a maximum.

9. A traffic processing method as in claim 8, wherein each of the two updating steps (f) and (i) includes the step of copying the entire content of the new destination field to the current destination field.

10. A traffic processing method as in claim 8, wherein each of the two updating steps (g) and (h) includes the step of copying the value of the corresponding destination bit in the new destination field to the destination bit in the current destination field.

11. A traffic processing method as in claim 1, wherein multicast traffic is provided better treatment than unicast traffic in terms of the location of memory address attachment in the linked-list of cell address, by further including the steps of:

a) appending the cell address of a multicast cell to a predetermined linked-list of cell addresses after a location pointed to by a corresponding multicast end pointer, if at the time of appending the cell address the number of cell addresses in the linked-list is equal to or exceeds a multicast attach threshold;

b) appending the cell address of a multicast cell to a predetermined linked-list of cell addresses after the location pointed to by the corresponding end pointer, if at the time of appending the cell address the number of cell addresses in the linked-list is less than the multicast attach threshold;

c) appending the cell address of a unicast cell to a predetermined linked-list of cell addresses after the location pointed to by the corresponding end pointer irrespective of the number of cell addresses in the linked-list.

12. A traffic processing method as in claim 1, wherein said output port is identified by finding the most significant bit in the destination field, which is located in the cell header, whose value is "1".

13. A traffic processing method as in claim 1, wherein said output port is identified by finding the least significant bit in the destination field, which is located in the cell header, whose value is "1".

14. A traffic processing method as in claim 1, wherein the contents of the linked-list of cell addresses are read one at a time according to a direction set by a traffic scheduler.

****

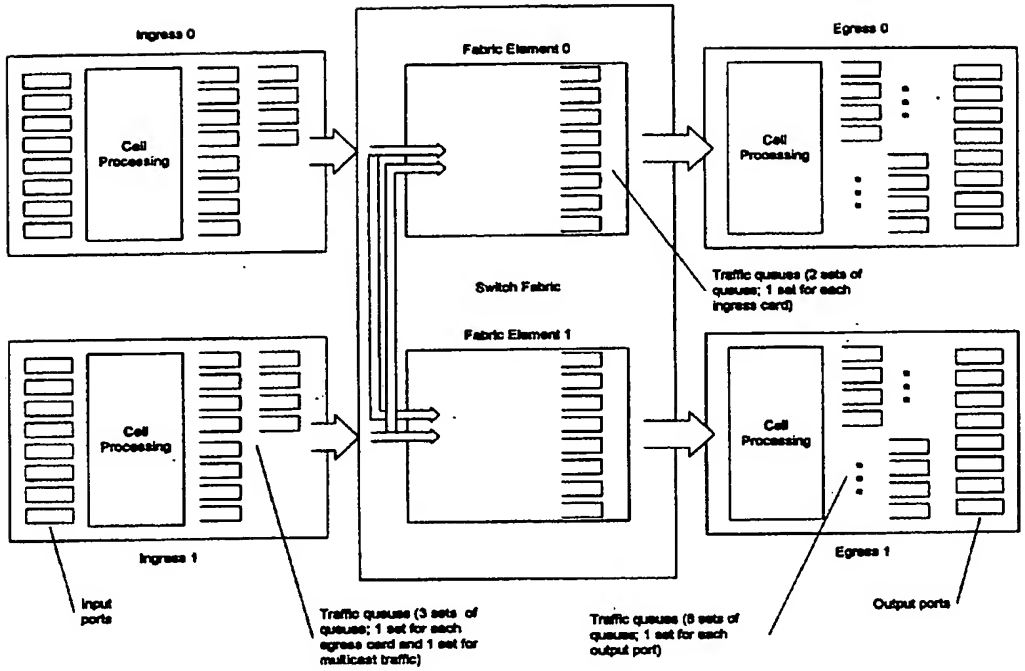


Figure 1

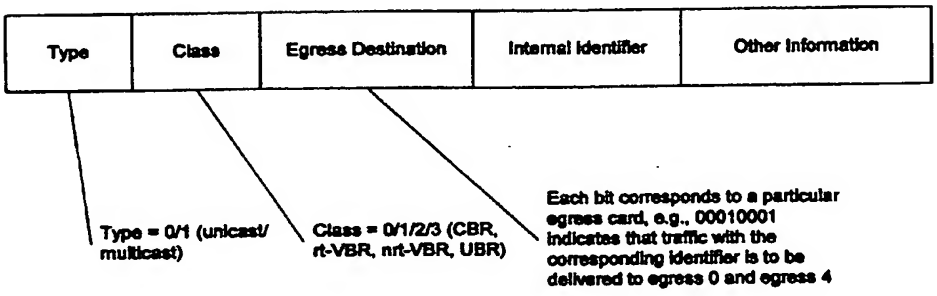


Figure 2

Borden Elliot Scott & Aylen

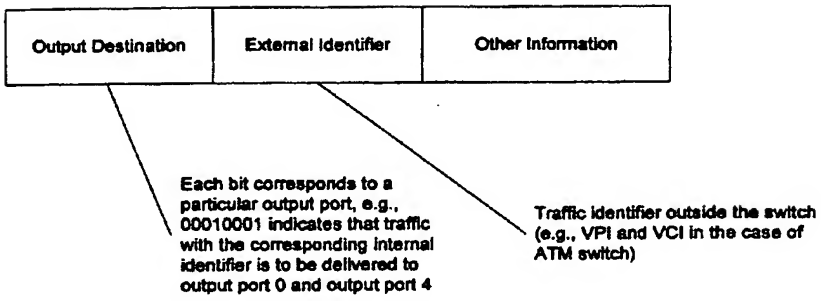


Figure 3

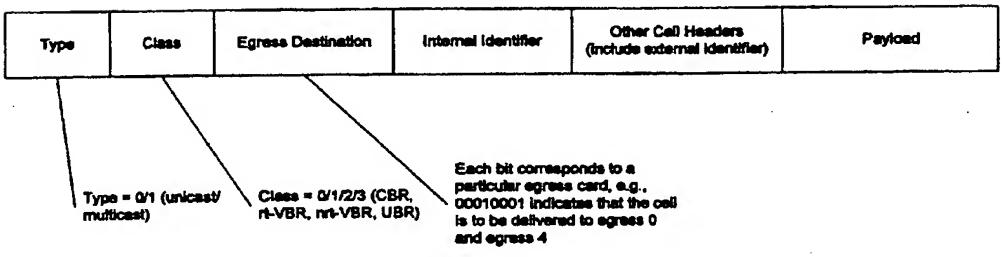


Figure 4

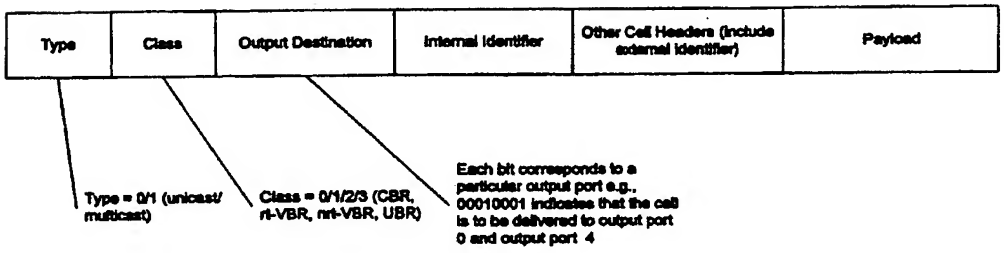


Figure 5

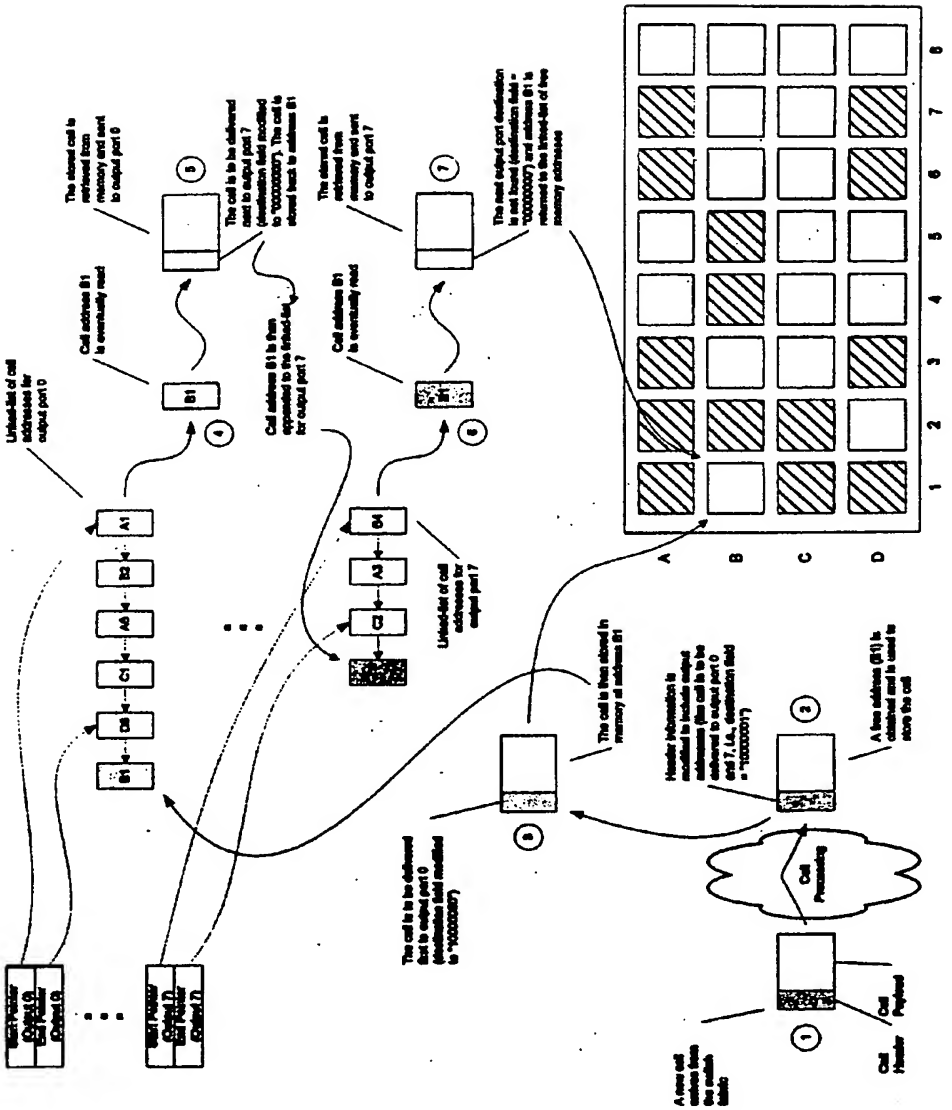


Figure 6

Borden Elliot Scott & Aylen

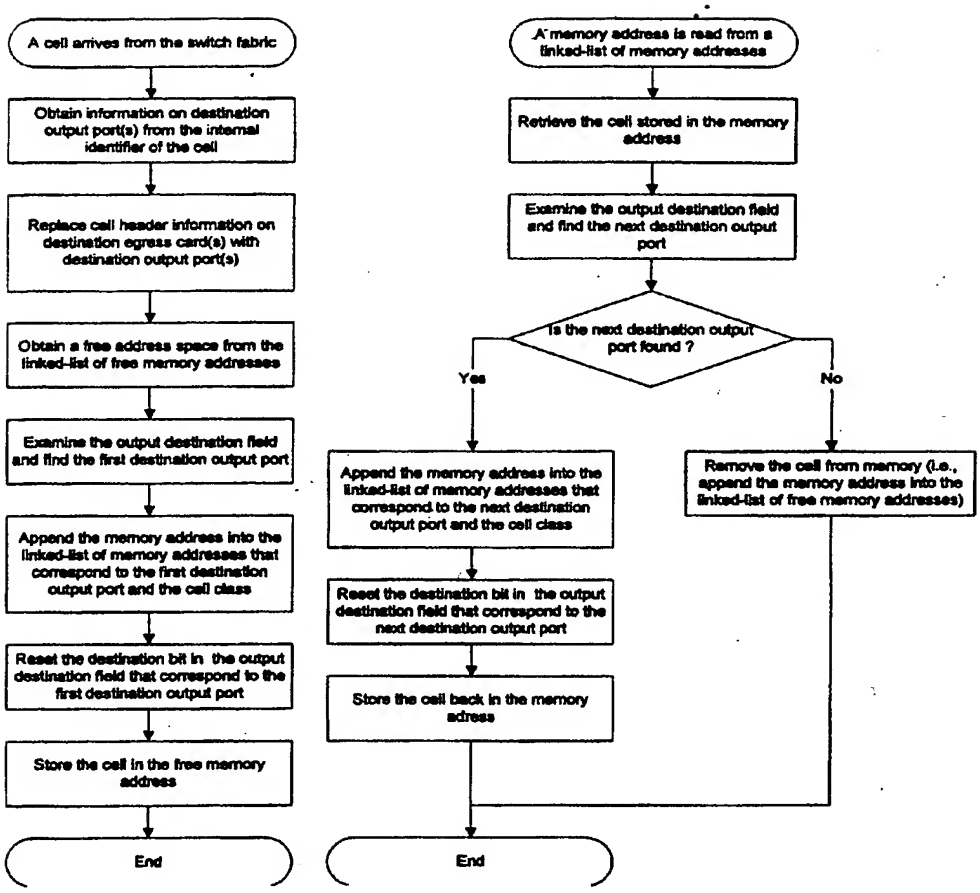


Figure 7

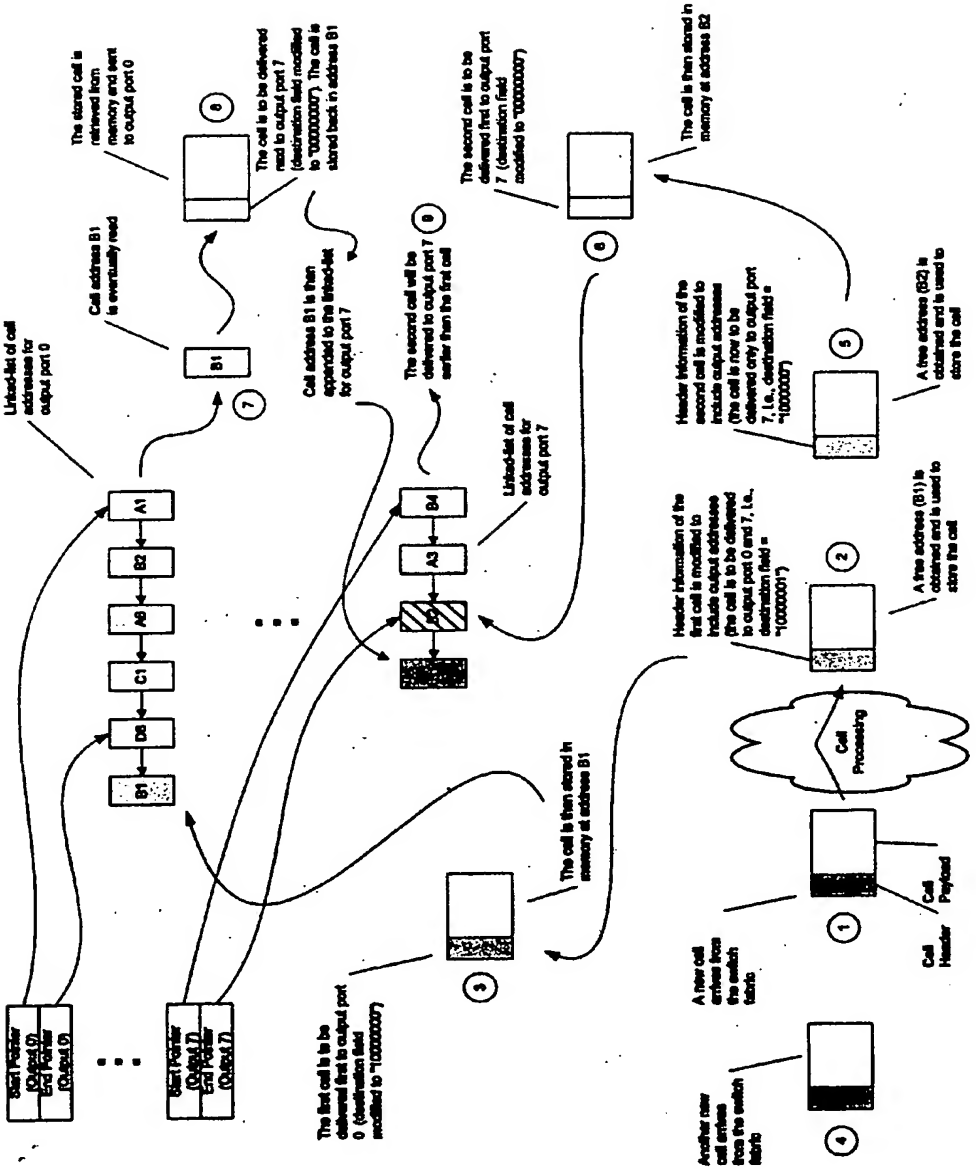


Figure 8

Borden Elliot Scott & Aylen

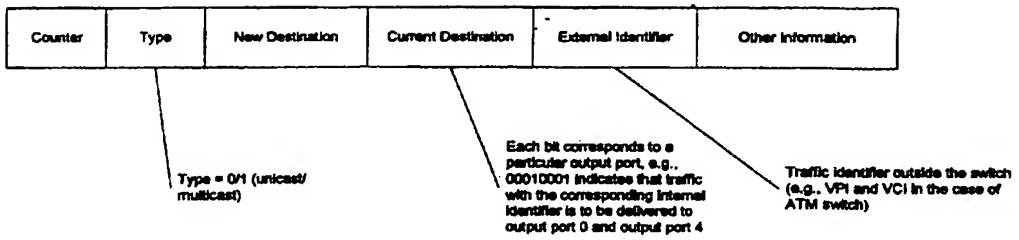


Figure 9

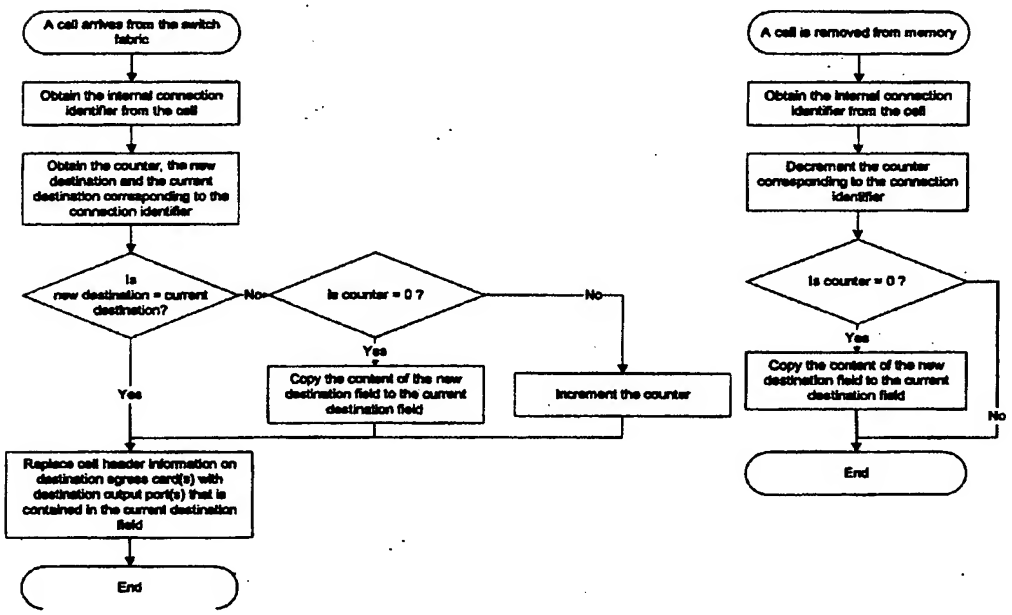


Figure 10

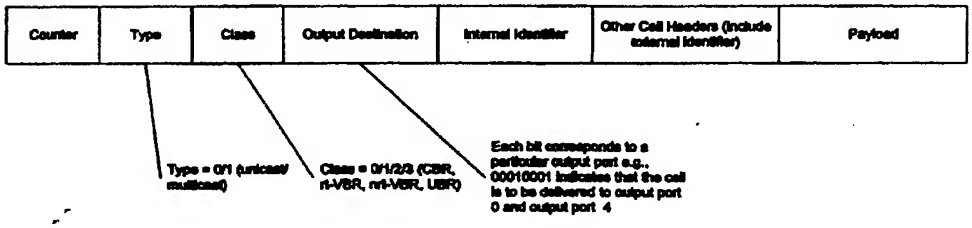


Figure 11

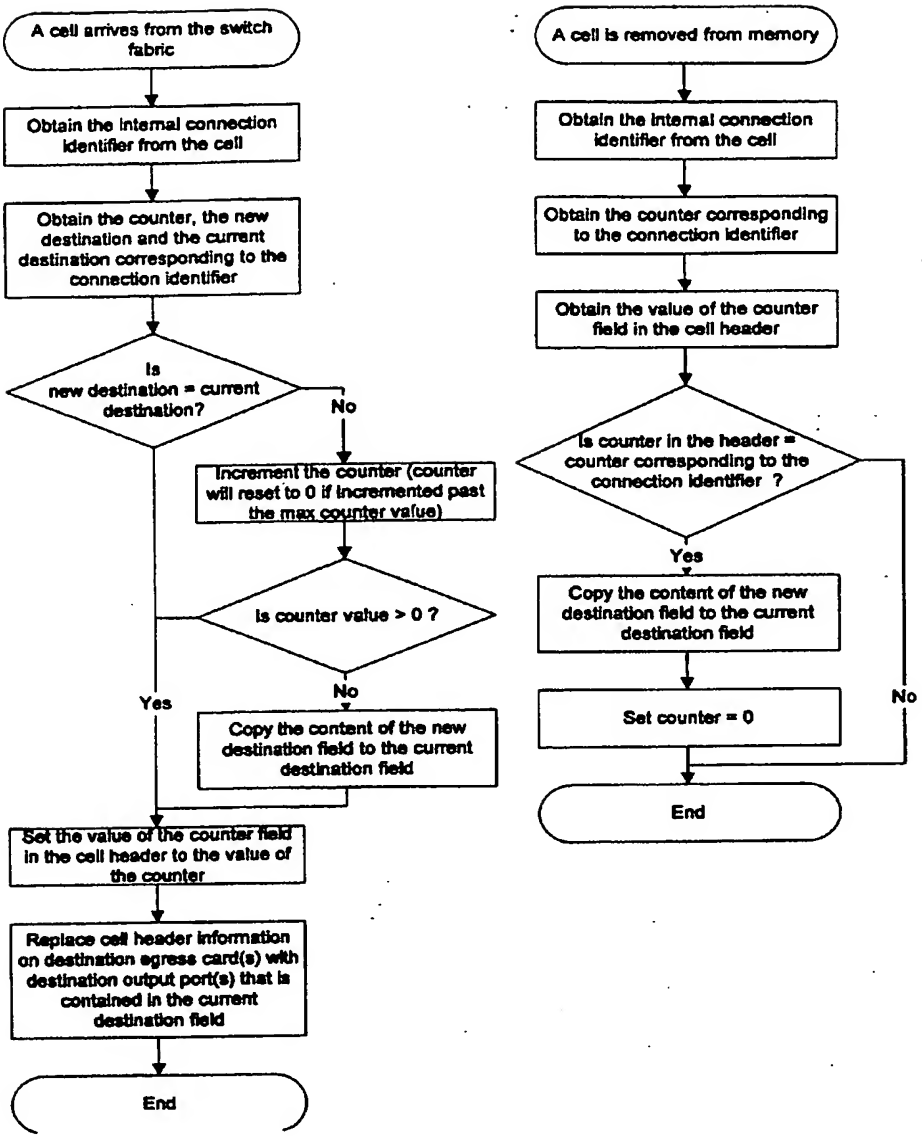


Figure 12

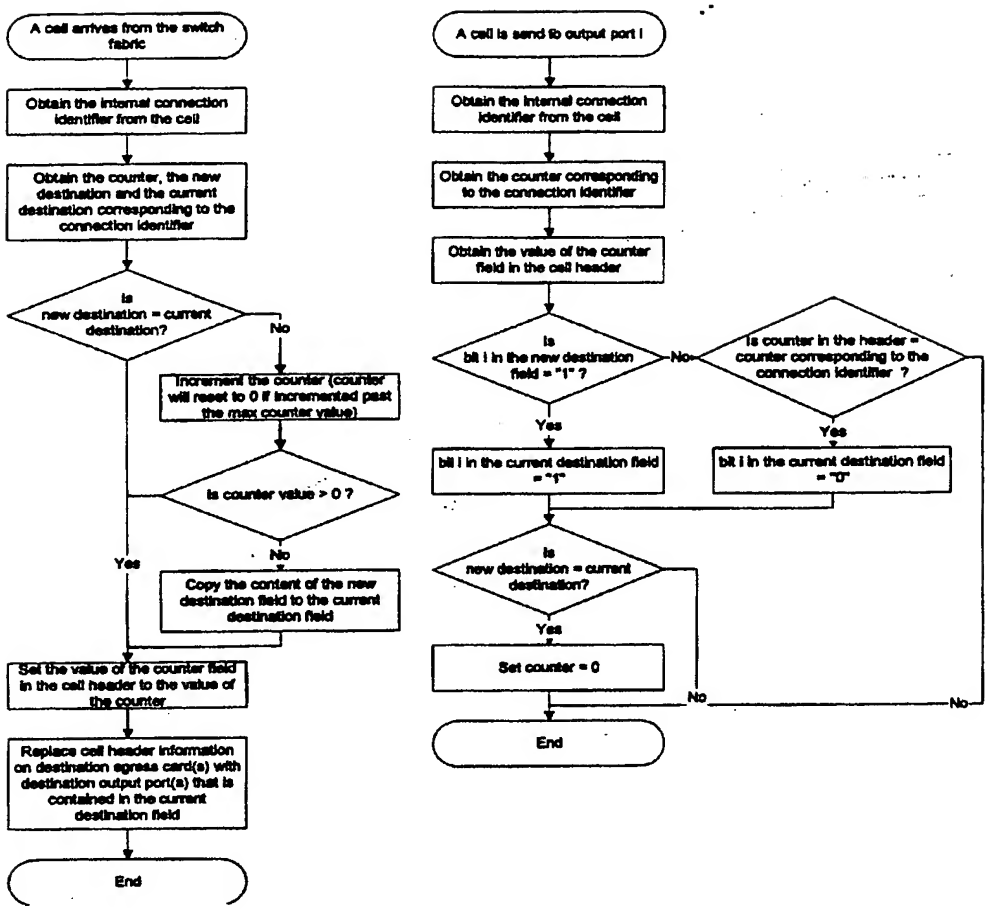


Figure 13

Borden Elliot Scott & Aylen

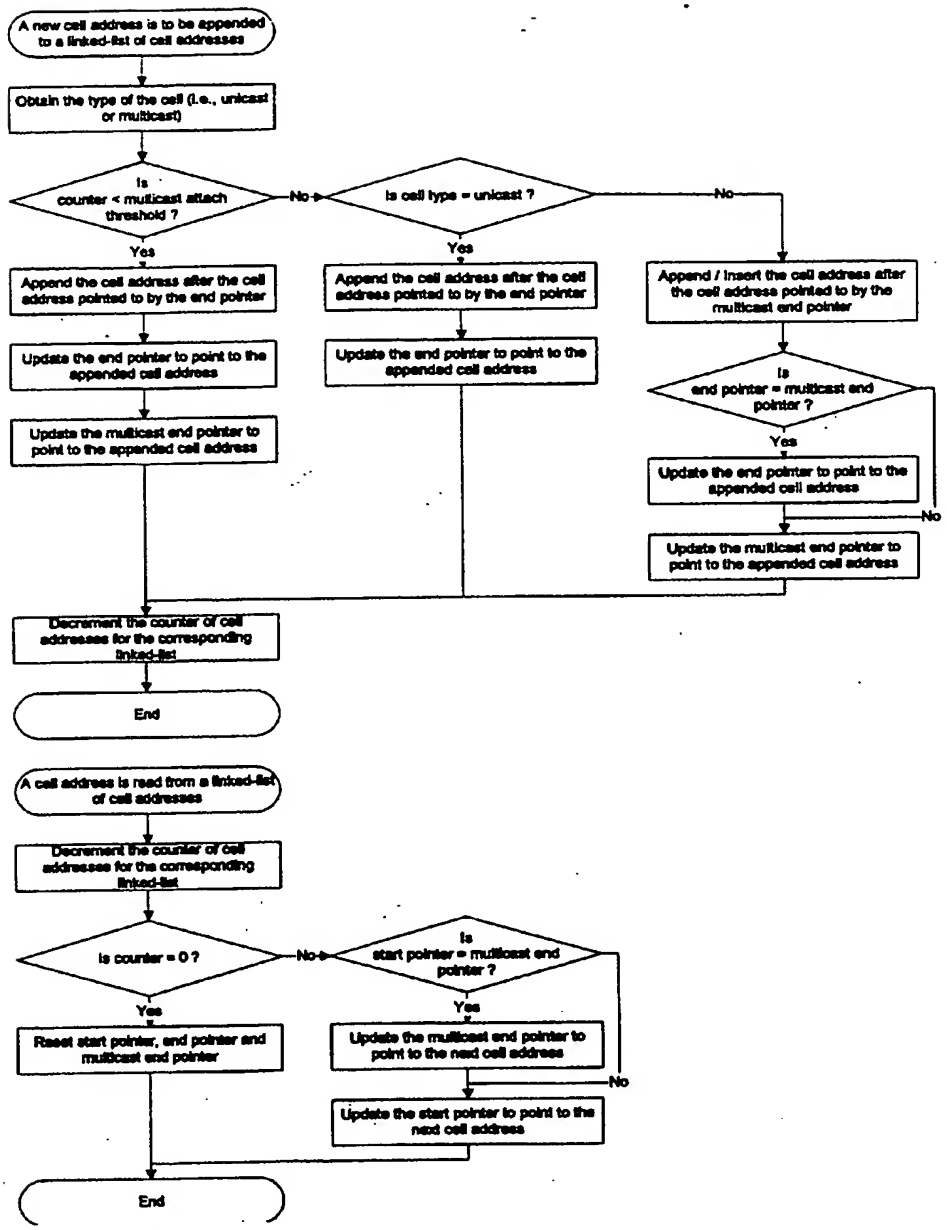


Figure 14

1	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	2
2	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	2
3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	2
4	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	2
5	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	2
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2
7	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	4
8	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	4
9	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	4
10	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	4
11	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	4
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	4
13	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	8
14	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	8
15	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	8
16	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	8
17	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	8
18	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	8

Figure 15

1	46	0.50
2	57	0.625
4	68	0.75
4	79	0.875
5	84	0.925

Figure 16

1	38	0.25	0.50
2	57	0.38	0.63
3	75	0.50	0.75
4	87	0.58	0.83
5	94	0.625	0.875

Figure 17

1	6	0.25	0.50
2	9	0.375	0.625
3	12	0.50	0.75
4	14	0.58	0.83
5	15	0.625	0.875

Figure 18

CBR	64	32	1
rt-VBR	64	32	1

Figure 19

CBR	1280	128	256	0
rt-VBR[12]	1280	128	256	0
rt-VBR[13]	5120	512	1024	1

Figure 20

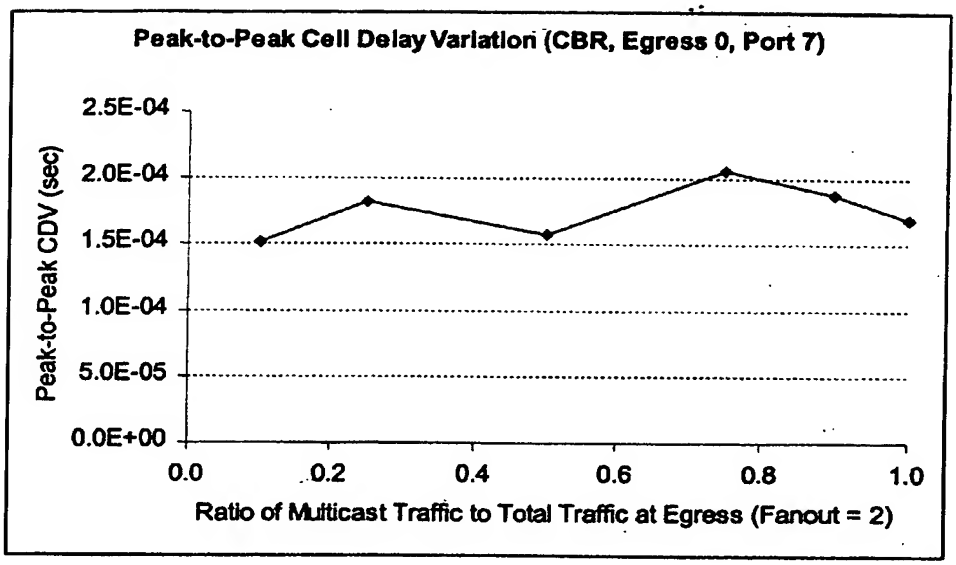


Figure 21

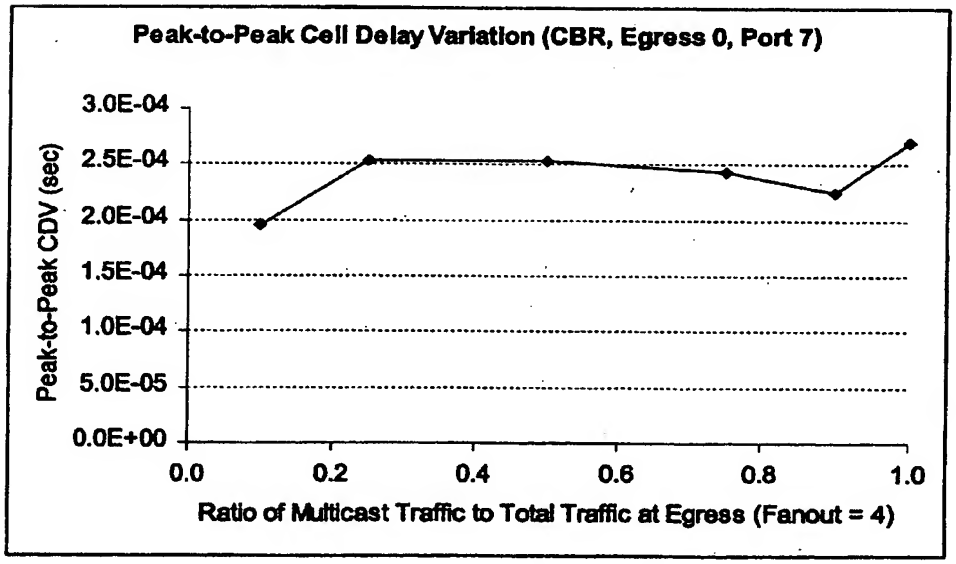


Figure 22

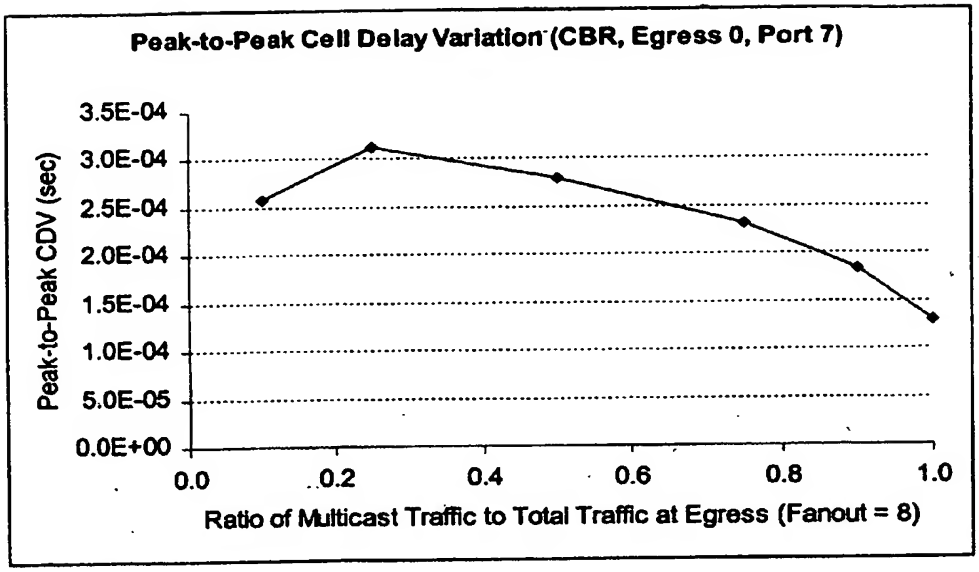


Figure 23

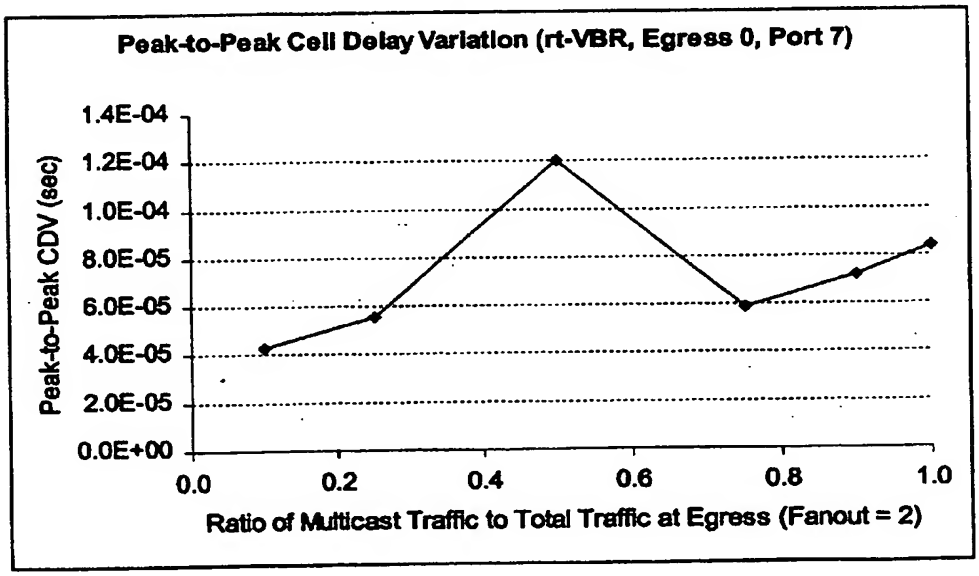


Figure 24

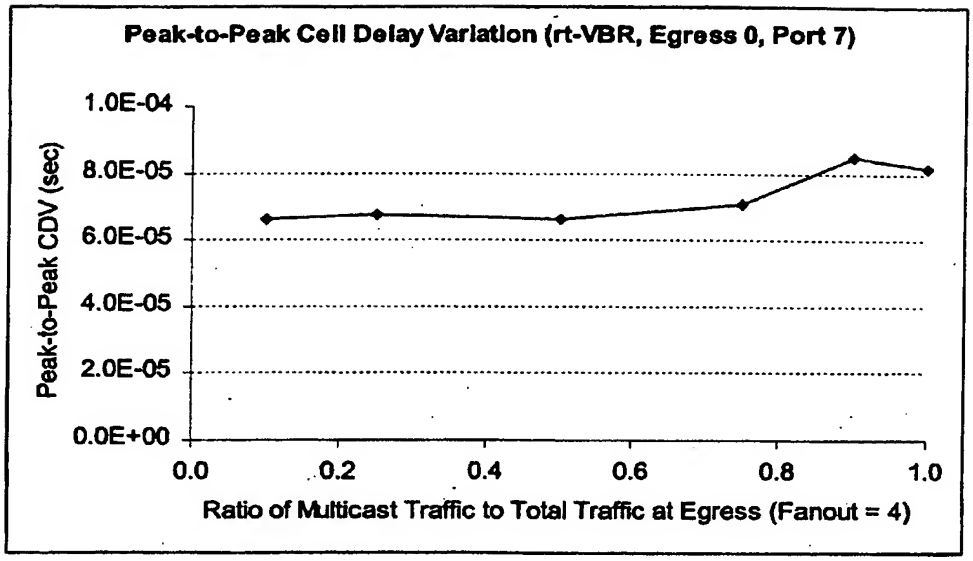


Figure 25

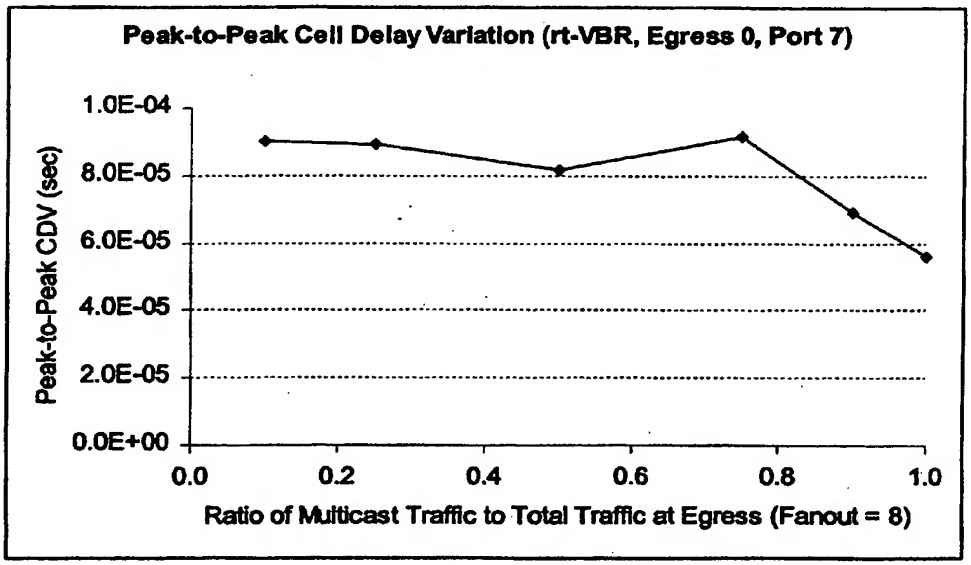


Figure 26

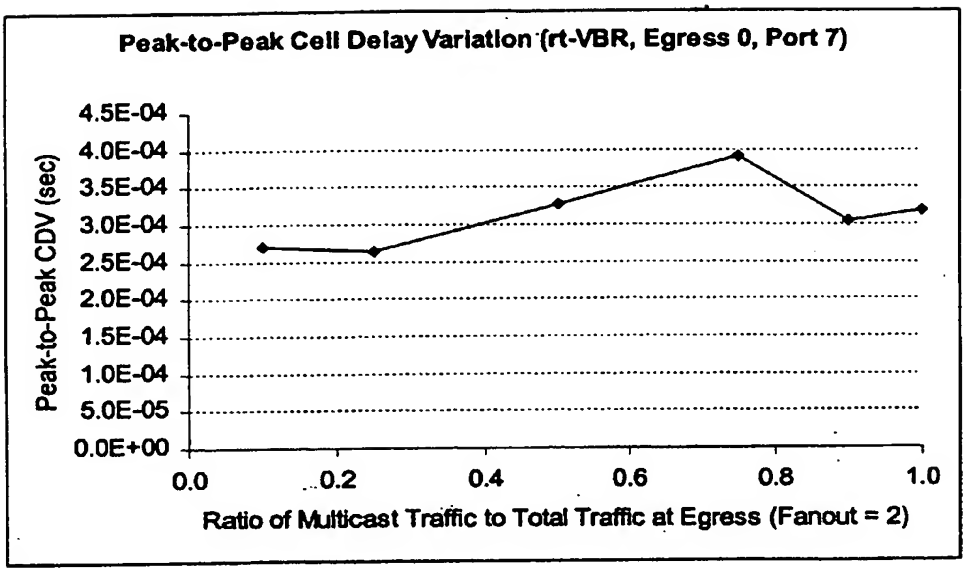


Figure 27

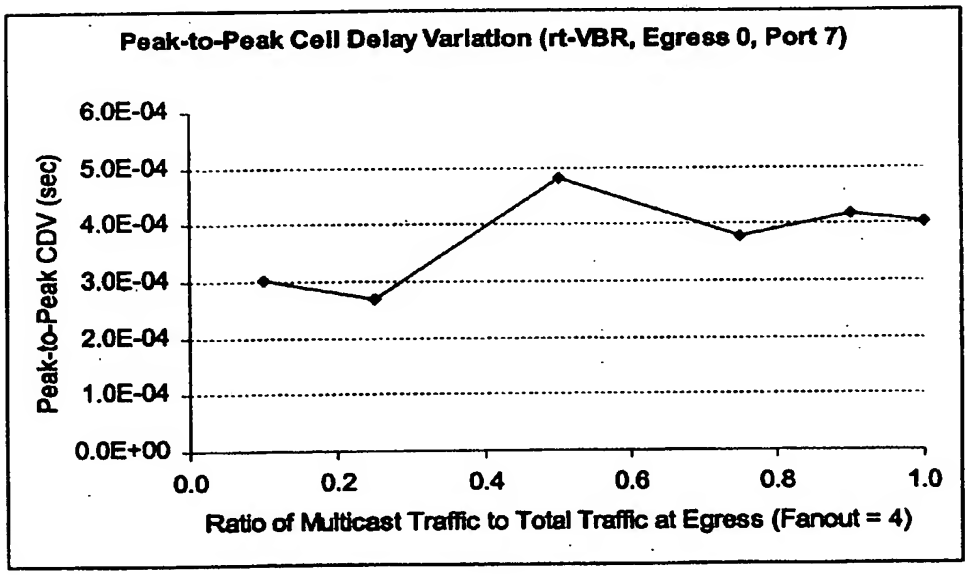


Figure 28

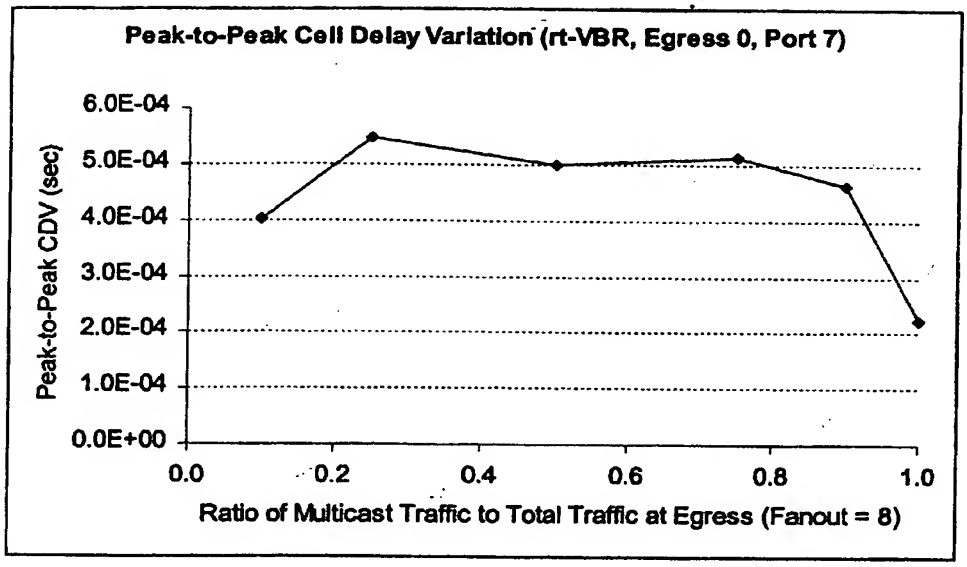


Figure 29

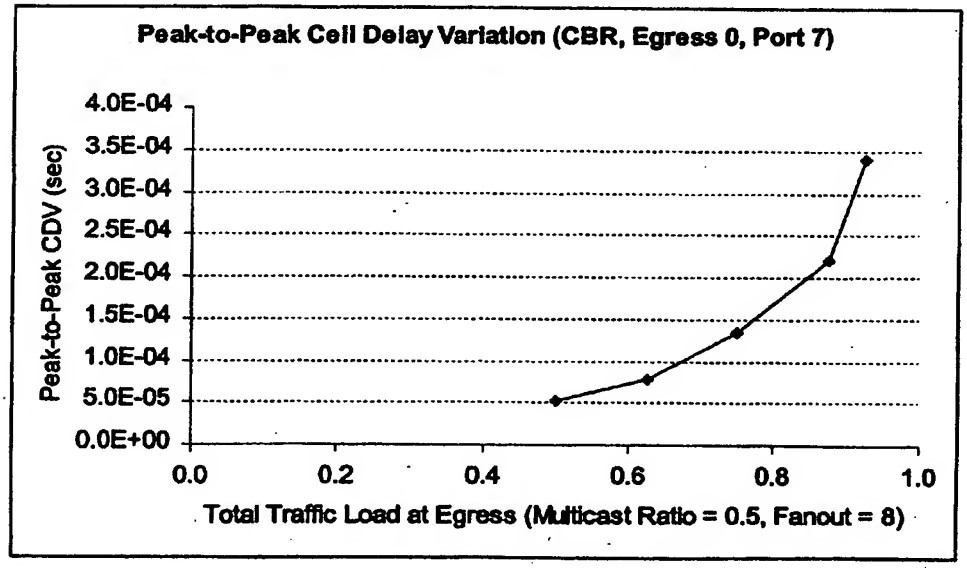


Figure 30

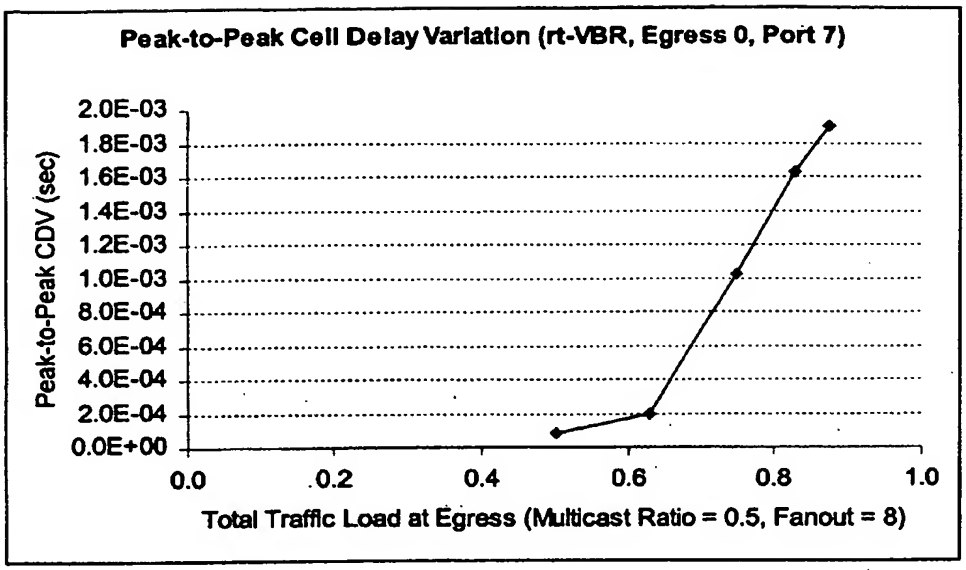


Figure 31

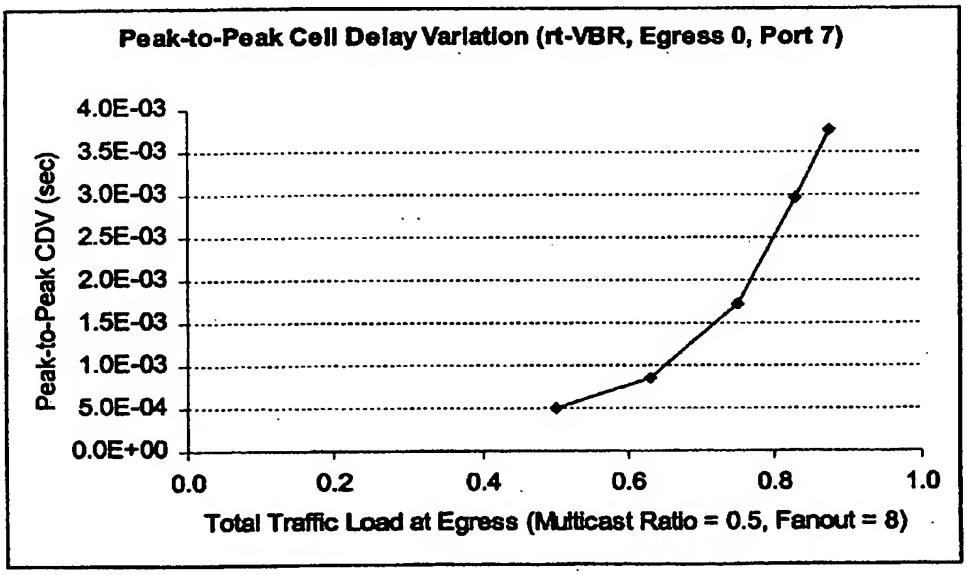


Figure 32